

# Aplikasi Graf dan Algoritma Dijkstra Dalam Jalur Penerbangan

Gama Pradipta Wirawan -13517049  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13517049@std.itb.ac.id

**Abstrak**—Saat ini pesawat terbang sudah menjadi alat transportasi umum yang sering digunakan oleh masyarakat khususnya Indonesia, baik digunakan untuk berpergian ke luar negeri atau digunakan untuk berpergian ke tempat yang tidak dapat dijangkau melalui transportasi darat. Karena banyaknya jumlah penumpang pesawat yang kian meningkat, dibutuhkan suatu cara agar transportasi ini dapat mencapai tujuan dengan rute yang sependek mungkin dan secepat mungkin. Namun tidak semua pesawat dibuat dengan kapasitas bahan bakar yang dapat langsung terbang dari satu tempat ke tempat lain tanpa harus melakukan transit terlebih dahulu. Oleh karena itu algoritma dijkstra ini digunakan untuk mencari rute se-efisien mungkin dengan bahan bakar yang terbatas.

**Kata Kunci**—Graf, Dijkstra, Rute, Transit.

## I. PENDAHULUAN

Musim liburan sudah hampir tiba dan kini telah menuju penghujung tahun dimana saat-saat ini adalah saat-saat yang membahagiakan bagi semua umat manusia karena kita akan segera merayakan Tahun Baru Masehi 2019 dan juga merayakan Natal. Di masa ini adalah waktu yang spesial bagi kita untuk berkumpul dengan keluarga kita dan merencanakan perjalanan bersama-sama sebagai suatu keluarga yang ceria. Banyak orang-orang yang juga rela menghabiskan uangnya untuk berjalanan ke tempat-tempat wisata yang indah. Bahkan banyak juga orang yang merelakan uangnya untuk membeli tiket pesawat untuk pergi keluar negeri bersama keluarganya. Dan apabila kita berbicara dengan acara pergi keluar negeri kita pasti takkan asing dengan kata-kata transit.



Gambar 1. Dubai On New Year's Eve (sumber: <https://novotel.accorhotels.com/gb/editorial/article/dubai-on-new->

[years-eve-best-places-to-watch-the-fireworks-this-year-3-15001-v7419.shtml](https://www.pinterest.com/pin/301459768783788351/))

Transit adalah perhentian sementara untuk berganti pesawat (atau melanjutkan pesawat yang sama) untuk melanjutkan perjalanan ke tempat tujuan. Adanya sistem transit disebabkan oleh beberapa faktor salah satunya adalah kapasitas bahan bakar pesawat yang terbatas.



Gambar 2. Perbandingan besar pesawat (sumber: <https://www.pinterest.com/pin/301459768783788351/>)

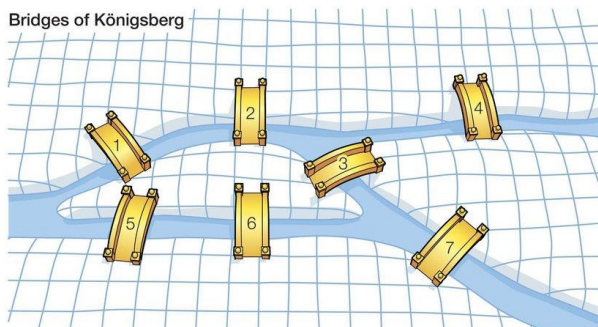
Jenis-jenis pesawat komersial di dunia ini sangatlah beragam. Ada pesawat komersial yang memang dirancang untuk melakukan perjalanan yang cukup jauh namun tidak jarang juga kita menemukan pesawat komersial yang kapasitas bahan bakarnya lebih kecil. Tapi ini bukan berarti pesawat kecil tidak dapat digunakan untuk perjalanan jauh. Pesawat kecil ini tetap dapat melakukan perjalanan jauh dengan sistem transit, yaitu berhenti di bandara tertentu untuk mengisi ulang bahan bakarnya. Namun maskapai penerbangan tidak mungkin mengatur rute penerbangan dengan asal-asalan. Mereka pasti tetap memperhitungkan cara agar pesawat tersebut mendapatkan

rute penerbangan se-efisien mungkin dan salah satu cara untuk mencari rute yang efisien adalah dengan menggunakan implementasi graf dan juga algoritma Dijkstra.

## II. LANDASAN TEORI

### 2.1 Graf

Graf dapat didefinisikan suatu pasangan himpunan yang terdiri dari himpunan simpul (*node*) dan himpunan sisi (*edge*) yang menghubungkan antar simpul. Sejarah teori mengenai graf bermula saat seorang ahli matematika Swiss yang bernama Leonhard Euler berhasil memecahkan masalah jembatan Königsberg. Jembatan Königsberg merupakan teka-teki mengenai kemungkinan untuk menemukan jalan setapak di tujuh jembatan yang membentang sepanjang sungai bercabang.



Gambar 1. Jembatan Königsberg (sumber: [https://www.researchgate.net/figure/The-Seven-Bridges-of-Koenigsberg\\_fig1\\_264707809](https://www.researchgate.net/figure/The-Seven-Bridges-of-Koenigsberg_fig1_264707809))

Dalam pengaplikasiannya biasanya untuk suatu graf  $G$  digunakan notasi matematis yang dinyatakan sebagai  $G = (V, E)$  dengan  $G$  adalah graf,  $V$  adalah simpul (*node*) dan  $E$  adalah sisi atau (*edge*).

Simpul pada graf dapat diberikan tanda atau nama berupa angka, karakter atau keduanya. Apabila terdapat suatu sisi ( $e$ ) yang menghubungkan dua simpul  $a$  dan  $b$ , maka sisi tersebut dapat ditulis sebagai :

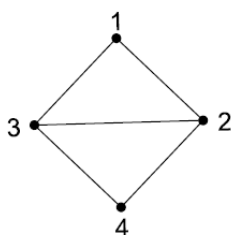
$$e = (a, b)$$

Graf dapat dikelompokkan menjadi beberapa jenis berdasarkan dasar pengelompokan yang berbeda-beda. Dasar pengelompokan itu adalah antara lain ada atau tidaknya gelang atau sisi ganda dan berdasarkan orientasi arah sisi pada graf.

Berdasarkan ada atau tidaknya gelang atau sisi ganda :

#### a) Graf Sederhana

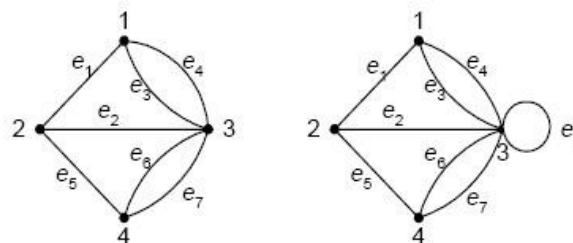
Graf sederhana merupakan graf yang tidak mengandung gelang maupun sisi-ganda. Contoh graf sederhana sebagai berikut :



Gambar 2. Graf Sederhana (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

#### b) Graf Tak Sederhana

Graf Tak sederhana merupakan graf yang mengandung sisi ganda atau gelang. Sisi gelang merupakan sisi yang berawal dan berakhir pada simpul yang sama. Sedangkan sisi ganda merupakan 2 atau lebih sisi yang berawal di tempat yang sama dan berakhir di tempat yang bersamaan. Graf yang mengandung graf ganda biasa disebut dengan Graf Ganda dan graf yang mengandung gelang biasa juga disebut sebagai Graf Semu. Contoh graf tak sederhana sebagai berikut :



Gambar 3. Graf Ganda (kiri) dan Graf Semu (kanan) (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

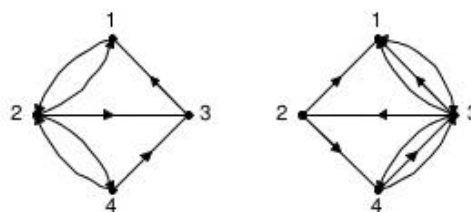
Berdasarkan orientasi arah sisi pada graf :

#### a) Graf tak-berarah

Graf tak-berarah merupakan graf yang sisinya tidak mempunyai orientasi arah. Pada kasus ini penulisan urutan pasangan simpul tidak berpengaruh sehingga sisi  $(a, b)$  dan  $(b, a)$  dianggap sama. Gambar 2 merupakan salah satu contoh graf tak-berarah.

#### b) Graf berarah

Graf berarah merupakan graf yang setiap sisinya diberikan orientasi arah. Pada kasus graf berarah ini penulisan urutan pasangan simpul pada sisi relevan. Berbeda dengan graf tak-berarah sisi  $(a, b)$  dan  $(b, a)$  memiliki arti yang berbeda. Contoh graf berarah adalah sebagai berikut :



Gambar 4. Graf Berarah (kiri) dan Graf Ganda Berarah(kanan) (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

Graf memiliki beberapa terminologi/ istilah yang perlu dipahami antara lain:

#### a) Ketetanggaan (*Adjacent*)

Dua buah simpul dikatakan bertetangga apabila

keduanya terhubung langsung. Pada graf tak-berarah dua buah simpul yang dikatakan bertetangga apabila terhubung oleh sisi. Sedangkan pada graf berarah dua simpul dikatakan bertetangga apabila terhubung oleh busur.

b) Bersisian (Incidency)

Sebuah sisi dikatakan bersisian dengan kedua sisi yang dihubungkannya. Dengan kata lain, apabila terdapat sebuah sisi  $e = (a,b)$  dapat dikatakan bahwa  $e$  bersisian dengan simpul  $a$  atau  $e$  bersisian dengan simpul  $b$ .

c) Simpul Terpencil (Isolated Vertex)

Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya. Simpul terpencil juga dapat dinyatakan sebagai sebuah simpul yang tidak bertetangga dengan simpul lainnya.

d) Graf Kosong (null graph atau empty graph)

Graf dapat dikatakan graf kosong apabila himpunan sisinya merupakan himpunan kosong. Dengan kata lain graf tersebut tak memiliki satupun sisi yang menghubungkan antara simpul-simpulnya. Simpul-simpul ini juga dapat disebut dengan simpul terpencil.

e) Derajat (Degree)

Derajat pada suatu simpul pada graf menunjukkan jumlah sisi yang bersisian dengan simpul tersebut. Derajat sebuah simpul  $A$  dapat dinyatakan sebagai  $d(A)$ . Sebuah simpul dikatakan sebagai simpul terpencil apabila derajat dari simpul tersebut adalah 0. Pada graf-semu, gelang pada suatu simpul dapat dihitung berderajat dua.

f) Lintasan (Path)

Lintasan yang panjangnya  $n$  dari simpul awal  $A$  ke simpul tujuan  $B$  dalam suatu graf adalah barisan simpul-simpul dan sisi-sisi yang menunjukkan arah dari simpul awal  $A$  ke simpul tujuan  $B$  yang melewati beberapa simpul dan sisi sebanyak  $n$ . Pada graf terdapat dua jenis lintasan yaitu :

~ Lintasan Euler

Lintasan Euler merupakan lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali.

~ Lintasan Hamilton

Lintasan Hamilton merupakan lintasan yang melalui masing-masing simpul di dalam graf tepat satu kali.

g) Siklus (Cycle) atau Sirkuit (Circuit)

Siklus atau sirkuit adalah suatu lintasan yang berawal dan berakhir pada simpul yang sama tanpa melewati simpul ataupun sisi yang telah dilewati sebelumnya. Pada graf terdapat dua jenis sirkus atau sirkuit yaitu :

~ Sirkuit Euler

Sirkuit Euler merupakan sirkuit yang melalui masing-masing sisi di graf tepat satu kali serta berakhir di simpul yang sama dengan simpul awal.

~ Sirkuit Hamilton

Sirkuit Hamilton merupakan sirkuit yang melalui masing-masing simpul di dalam graf tepat satu kali serta berakhir di simpul yang sama dengan simpul awal.

h) Terhubung (Connected)

Dua buah simpul  $a$  dan  $b$  disebut terhubung jika terdapat lintasan dari  $a$  ke  $b$ . Suatu graf dapat dikatakan graf terhubung apabila untuk setiap pasangan simpul-simpulnya  $a$  dan  $b$  terdapat lintasan dari  $a$  ke  $b$ . Jika tidak maka graf tersebut dikatakan graf tak-terhubung.

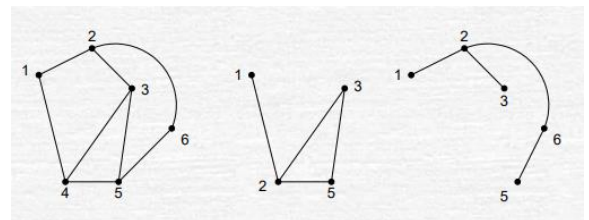
Dua simpul  $a$  dan  $b$  pada graf berarah  $G$  disebut terhubung kuat apabila terdapat lintasan berarah dari  $a$  ke  $b$  dan juga terdapat lintasan berarah dari  $b$  ke  $a$ . Sedangkan jika  $a$  dan  $b$  tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya maka  $a$  dan  $b$  dikatakan terhubung lemah.

Suatu Graf berarah  $G$  dikatakan graf terhubung kuat apabila untuk setiap pasang simpul sembarang  $a$  dan  $b$  di  $G$ , terhubung kuat. Kalau hal ini tidak terpenuhi maka graf  $G$  ini disebut graf terhubung lemah.

i) Upagraf (Subgraph) dan Komplemen Upagraf

Misalkan  $G = (V,E)$  adalah sebuah graf.  $G_1 = (V_1,E_1)$  adalah upagraf dari  $G$  jika  $V_1 \subseteq V$  dan  $E_1 \subseteq E$ .

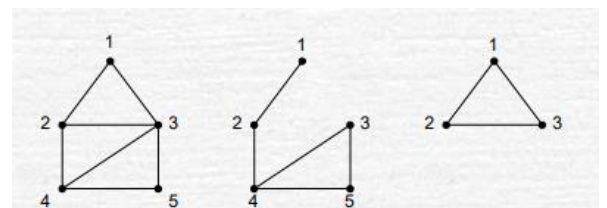
Komplemen dari upagraf  $G_1$  terhadap graf  $G$  adalah graf  $G_2 = (V_2,E_2)$  sedemikian sehingga  $E_2 = E - E_1$  dan  $V_2$  adalah himpunan simpul yang anggota-anggota  $E_2$  bersisian dengannya.



Gambar 5. Graf  $G$ (kiri), Upagraf  $G_1$ (tengah), dan komplemennya (kanan). (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

j) Upagraf Rentang (Spanning Subgraph)

Upagraf  $G_1 = (V_1,E_1)$  dari  $G = (V,E)$  dikatakan upagraf rentang jika  $V_1 = V$  (yaitu  $G_1$  mengandung semua simpul dari  $G$ ).

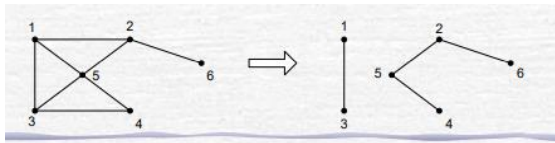


Gambar 6. Graf  $G$ (kiri), Upagraf rentang dari  $G$  (tengah), Bukan upagraf rentang dari  $G$  (kanan). (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))



k) Cut-Set

Cut-Set dari graf terhubung  $G$  adalah himpunan sisi yang bila dibuang dari  $G$  menyebabkan  $G$  tidak terhubung. Hal ini menyebabkan  $G$  akan menghasilkan dua buah komponen.



Gambar 6.  $G$  sebelum di cut-set(kiri),  $G$  setelah di cut-set(kiri)(sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

l) Graf Berbobot (Weighted Graph)

Sebuah graf berbobot adalah graf yang sisi-sisinya diberi sebuah bobot. Arti bobot ini berbeda-beda tergantung pada aplikasi dari graf. Dalam hal ini kita akan menggunakan graf berbobot untuk memodelkan waktu tempuh yang diperlukan untuk menuju tempat 1 ke tempat lain.

2.2 Algoritma Dijkstra

Algoritma Dijkstra merupakan sebuah algoritma yang diciptakan oleh Edsger Dijkstra yang digunakan untuk memecahkan permasalahan jarak terpendek untuk sebuah graf berarah (kali ini kita akan mempresentasikan bobot sebagai waktu bukan jarak). Secara garis besar algoritma ini bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke  $n$ , setidaknya ada  $n$  node yang sudah kita tahu jalur terpendek. Langkah-langkah algoritma Dijkstra ini dapat di dilakukan sebagai berikut :

1. Tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap.
2. Beri nilai bobot(jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain ( belum terisi).
3. Set semula node yang belum dilalui dan set node awal sebagai "Node keberangkatan"
4. Dari node keberangkatan, pertimbangkan node tetangga yang belum dilalui lalu hitunglah jarak dari titik keberangkatan. Apabila lebih kecil dari jarak sebelumnya maka hapus data lama dan simpan ulang data jarak dengan jarak yang baru.
5. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilalui sebagai "Node dilewati". Node yang dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan paling minimal bobotnya.
6. Set "Node belum dilewati" dengan jarak terkecil sebagai "Node Keberangkatan".

III. METODOLOGI

3.1 Pembuatan Graf

Pada saat ingin mempresentasikan bahan bakar yang dibutuhkan saat terbang, penulis dapat mempresentasikannya dengan graf. Dalam hal ini, penulis menggunakan graf berbobot untuk mempresentasikannya. Kita dapat menganggap simpul (*node*) sebagai kota dan sisi (*edge*) sebagai banyak bahan bakar yang dibutuhkan untuk melakukan rute penerbangan tersebut. Bobot-bobot yang terdapat di sisi dapat kita anggap menjadi bahan bakar minimal yang dibutuhkan dalam penerbangan itu. Satuan yang penulis pakai untuk bahan bakar pesawat ini adalah dalam satuan liter.

Pertama-tama yang kita harus lakukan adalah memberi batasan-batasan tertentu. Salah satunya adalah batasan transit dimana kebanyakan orang mungkin hanya ingin untuk melakukan transit sebanyak 1-2 kali. Sehingga kita hanya bisa mencari data berdasarkan kriteria-kriteria yang kita batasi. Dari cara tersebut, misalkan, kita mendapatkan data sebagai berikut. (Dalam hal ini, untuk bahan bakar dari B ke A dan A ke B sama).

Asal	Tujuan	Bahan Bakar
A	B	100
A	C	111
A	D	143
B	A	100
B	C	55
B	D	87
C	A	111
C	B	55
C	D	70
D	A	143
D	B	87
D	C	70

Tabel 1. Tabel data bahan bakar yang dibutuhkan dari kota 1 ke kota lain

Kemudian apabila kita sudah mengumpulkan data bahan bakar yang dibutuhkan, kita dapat melanjutkan ke proses berikutnya yaitu penghilangan rute yang bahan bakarnya melebihi kapasitas bahan bakar maksimal dari pesawat. Misalkan kapasitas bahan bakar maksimal dari pesawat X adalah 120, maka semua penerbangan yang membutuhkan bahan bakar lebih banyak dari 120 akan dihapus, sehingga menghasilkan tabel data seperti berikut ini.

Asal	Tujuan	Bahan Bakar
A	B	100
A	C	111
A	D	143
B	A	100
B	C	55
B	D	87
C	A	111
C	B	55
C	D	70
D	A	143
D	B	87

D	C	70
---	---	----

Tabel 2. Tabel data bahan bakar yang dibutuhkan dari kota 1 ke kota lain (Tanda merah berarti rute penerbangan tidak dapat dilakukan)

### 3.2 Modifikasi Algoritma

Dalam mencari rute penerbangan yang efisien dalam hal bahan bakar kita perlu menggunakan algoritma Dijkstra. Langkah-langkah yang perlu kita lakukan adalah sebagai berikut.

1. Pilih salah satu kota yang menjadi titik awal penerbangan dan pilih kota yang akan menjadi titik akhir penerbangan.
2. Apabila terdapat sisi yang menghubungkan titik awal dan titik akhir maka ambil sisi tersebut dan masukkan kedalam data simpanan.
3. Pilih sisi lain terkecil yang belum pernah disinggahi dan masukkan dalam data.
4. Ulangi langkah 3 dengan *first city* berubah menjadi *current city* hingga mencapai kota yang diinginkan.
5. Apabila menemui kota yang diinginkan maka jumlahkan bahan bakar yang dibutuhkan dan periksa apakah jumlah ini lebih kecil dari data simpanan. Apabila data bahan bakar yang baru lebih sedikit maka gantilah data bahan bakar yang lama dengan yang baru.
6. Ulangi langkah 3 sampai 5 namun sisi yang diambil adalah sisi yang lebih besar dari pada sisi yang sebelumnya diambil.
7. Ulangi langkah 3 sampai 5 kembali namun saat mencari rute 2 gunakan sisi yang bobotnya lebih besar dari rute 1.
8. Ulangi langkah-langkah tersebut namun ketika sudah mengulang sebanyak (n+1)-kali maka proses dihentikan, karena n adalah maksimal transit yang boleh dilakukan.

## IV. STUDI KASUS

### 4.1 Studi Kasus 1

Misalkan Pesawat X merupakan pesawat kecil yang memiliki kapasitas bahan bakar yang cukup kecil yaitu 120 liter. Rute penerbangan yang akan dilakukan adalah penerbangan dari kota A ke kota D. Berdasarkan pada data kita (Tabel 1) maka kita dapat menyimpulkan bahwa pesawat X tidak dapat melakukan penerbangan langsung dari A ke D. Oleh karena itu kita mengubah tabel data kita menjadi data baru seperti berikut ini.

Asal	Tujuan	Bahan Bakar
A	B	100
A	C	111
A	D	143
B	A	100
B	C	55
B	D	87
C	A	111
C	B	55
C	D	70
D	A	143
D	B	87
D	C	70

Tabel 3. Tabel data bahan bakar yang dibutuhkan dari kota 1 ke

kota lain untuk pesawat X (Tanda merah berarti rute penerbangan tidak dapat dilakukan)

Dari Tabel diatas kita dapat melakukan proses berikutnya yaitu pencarian rute terefektif yang bisa dilakukan oleh pesawat X. Proses pencariannya adalah sebagai berikut.

- ~ Apabila pesawat X mengambil jalur ke kota B lalu ke Kota D maka bahan bakar yang diperlukan adalah  $100+87 = 187$  liter.
- ~ Apabila pesawat X mengambil jalur ke kota B lalu ke Kota C lalu ke Kota D maka bahan bakar yang diperlukan adalah  $100+55+70 = 225$  liter.
- ~ Apabila pesawat X mengambil jalur ke kota C lalu ke Kota D maka bahan bakar yang diperlukan adalah  $111+70 = 181$  liter.
- ~ Apabila pesawat X mengambil jalur ke kota C lalu ke Kota B lalu ke Kota D maka bahan bakar yang diperlukan adalah  $111+55+87 = 253$  liter.

Berdasarkan data dari proses pencarian diatas maka dapat disimpulkan bahwa rute penerbangan pesawat X dari Kota A ke Kota B adalah A-C-D, dimana pesawat tersebut akan melakukan transit sebanyak 1 kali dan total bahan bakar yang dipakai dalam penerbangan ini adalah sebesar 181 liter.

### 4.2 Studi Kasus 2

Misalkan Pesawat Y merupakan pesawat besar yang memiliki kapasitas bahan bakar yang cukup besar yaitu 240 liter. Rute penerbangan yang akan dilakukan adalah penerbangan dari kota A ke kota D. Berdasarkan pada data kita (Tabel 1) maka kita dapat menyimpulkan bahwa pesawat Y dapat melakukan semua penerbangan yang ada di data tabel tersebut. Maka dalam kasus ini kita akan memakai data seperti berikut ini.

Asal	Tujuan	Bahan Bakar
A	B	100
A	C	111
A	D	143
B	A	100
B	C	55
B	D	87
C	A	111
C	B	55
C	D	70
D	A	143
D	B	87
D	C	70

Tabel 4. Tabel data bahan bakar yang dibutuhkan dari kota 1 ke kota lain untuk pesawat Y

Dari Tabel diatas kita dapat melakukan proses berikutnya yaitu pencarian rute terefektif yang bisa dilakukan oleh pesawat Y. Proses pencariannya adalah sebagai berikut.

- ~ Apabila pesawat Y mengambil jalur langsung ke Kota D maka bahan bakar yang diperlukan adalah 143 liter.
- ~ Apabila pesawat Y mengambil jalur ke kota B lalu ke Kota D maka bahan bakar yang diperlukan adalah  $100+87 = 187$  liter.
- ~ Apabila pesawat Y mengambil jalur ke kota B lalu ke Kota

- C lalu ke Kota D maka bahan bakar yang diperlukan adalah  $100+55+70 = 225$  liter.
- ~ Apabila pesawat Y mengambil jalur ke kota C lalu ke Kota D maka bahan bakar yang diperlukan adalah  $111+70 = 181$  liter.
  - ~ Apabila pesawat Y mengambil jalur ke kota C lalu ke Kota B lalu ke Kota D maka bahan bakar yang diperlukan adalah  $111+55+87 = 253$  liter.

Berdasarkan data dari proses pencarian diatas maka dapat disimpulkan bahwa rute penerbangan pesawat Y dari Kota A ke Kota B adalah A-D, dimana pesawat tersebut akan melakukan transit sebanyak 0 kali dan total bahan bakar yang dipakai dalam penerbangan ini adalah sebesar 143 liter.

## V. KESIMPULAN

Algoritma Dijkstra merupakan suatu algoritma turunan dari algoritma Prim yang berfungsi untuk mencari jarak efisien dari suatu simpul ke simpul yang lain pada graf. Pengaplikasian algoritma ini dapat dilakukan untuk melakukan pencarian rute efisien pesawat. Sehingga dapat disimpulkan bahwa teori aplikasi graf dan algoritma Dijkstra sangat berguna dalam membantu mencari rute penerbangan yang efisien dalam hal penggunaan bahan bakar.

## VII. UCAPAN TERIMA KASIH

Pertama-tama saya mengucapkan terima kasih kepada Tuhan Yesus Kristus, atas berkat dan kasih-Nya yang selalu melimpah dalam hidup saya sehingga makalah Matematika Diskrit ini dapat diselesaikan tepat waktu. Saya juga mengucapkan terima kasih kepada kedua orang tua saya yang selalu mendukung saya sehingga saya dapat menempuh pendidikan sampai saat ini. Tak lupa saya juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT. yang berperan sebagai dosen mata kuliah IF 2120 Matematika Diskrit sehingga dengan ilmu pengetahuan seputar Matematika Diskrit, saya dapat membuat dan menyelesaikan makalah ini.

## REFERENCES

- [1] Munir, Rinaldi. 2005. Matematika Diskrit, edisi 3. Bandung: Informatika Bandung.
- [2] <https://yuliee.wordpress.com/2010/04/22/pengertian-graph/>. Diakses pada tanggal 09 Desember 2018
- [3] <https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra/>. Diakses pada tanggal 09 Desember 2018
- [4] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf). Diakses pada tanggal 09 Desember 2018
- [5] [https://www.researchgate.net/figure/The-Seven-Bridges-of-Koenigsberg\\_fig1\\_264707809](https://www.researchgate.net/figure/The-Seven-Bridges-of-Koenigsberg_fig1_264707809). Diakses pada tanggal 09 Desember 2018
- [6] <https://novotel.accorhotels.com/gb/editorial/article/dubai-on-new-years-eve-best-places-to-watch-the-fireworks-this-year-3-15001-v7419.shtml>. Diakses pada tanggal 09 Desember 2018
- [7] <https://www.pinterest.com/pin/301459768783788351/>. Diakses pada tanggal 09 Desember 2018
- [8] <http://ranselkecil.com/tips/seputar-transit-dan-kegiatan-selama-transit/>. Diakses pada tanggal 09 Desember 2018

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2018



Gama Pradipta Wirawan -- 13517049